COLUMN

シリコンバレーに学ぶキャリア・デベロップメント

第13回

履歴書作成術



東原朋成

以前，本誌に連載していたコラムで履歴書（レジュメ）について簡単に触れたことがあります（本誌2004年1月号，pp.56-57のコラム「シリコンバレー就職事情 第8回『履歴書』」を参照）．米国では，キャリア・デベロップメント（キャリア開発）に履歴書は不可欠です．日本とは異なり，市販の履歴書を文房具店で購入することはありませんし，会社指定のフォーマットというのもありません（application form，すなわち申請書は会社ごとにあるが，これは履歴書ではない）．

履歴書は転職するときに書くものではなく，折を見てはアップデートし続けていく，自伝のようなものです．上述のコラムでも述べましたが，シリコンバレーではエンジニアやマネージャは履歴書のアップデートを欠かしません．転職するときだけでなく，プロジェクトの終了，昇進，配置換えが行われた時期，あるいは年末といったカレンダ上の区切りなどでアップデートを行っているようです．

「履歴書をアップデートする必要がないということは，何もしてこなかったということだ」と言い切るリクルータもいます．そういえば，リクルータは2年に一度は自分の手元にあるすべての履歴書について，アップデートする必要があるかどうかをチェックしていると聞きます（連絡が取れなくなった人の履歴書は破棄する）．履歴書は一人につき一つではありません．数種類の履歴書を作っている場合がほとんどです（例えば，エンジニアとしてのポジション用，マネージャとしてのポジション用など）．

それらを補足するために，より詳しい履歴書ノートを作成することも多いと聞きます．履歴書では「×××プロジェクトに参加」と書いてあっても，履歴書ノートには具体的に自分が何をしたのか，どのようなことがあったのかなどをメモしておくわけです．履歴書ノートは，転職インタビューなどで威力を発揮します．「自分がかかわったプロジェクトだから覚えている」と思っていても，1年もすると記憶は不確かになるものです．ですから，履歴書ノートを使って記憶を呼び起こしておけば，インタビューに役立つわけです．

現在，製薬関連会社の広報部門（投資家関係）のVice Presidentである友人は，年度末ごとに1日かけて履歴書と履歴書ノートをアップデートしているそうです．1年の区切りのイベントとして，20年もの間，これを行っているのだから，たいしたものです．

米国では，履歴書についていくつもの書籍が出版されています．また，インターネット上にも無数の関連記事（サイト）が公開されています．履歴書の書きかたに関するセミナや勉強会も頻繁に開催されています．内容としては，転職するのが初めてという人やビザで就労している（米国のカルチャにあまり通じていない）人など，転職初心者を対象としたいわゆる「ハウ・ツーもの」から，エグゼクティブなどを対象とした非常に高度なものまであるようです．

キャリア・コンサルタントとかかわるようになって最初に行うことは，履歴書（複数のバージョン）および履歴書ノートの作成だと聞いたことがあります．また，履歴書アドバイザと称する職業もあると耳にしました．

履歴書がすべての始まりです．ジョブ・インタビューにたどり着くには，まず履歴書を読んでもらわなくてはなりません．そのために，山ほどある履歴書の中から相手に選ばれる（相手に読ませる）ものを作成する必要があります．以前，話す技術（本誌2005年6月号，pp.118-119の本コラム第4回「プレゼンテーション能力」）や聞く技術（同年7月号，pp.92-93の本コラム第5回「聞く技術」）の必要性を述べましたが，読ませる技術も必要というわけです（読ませる技術とは，言い換えれば書く技術ということになる）．

だからといって，履歴書を誇大に記述したり，ウソの記述を行ってはなりません．履歴書を誇大表現するケースは後を絶ちませんが，履歴書を見慣れたマネージャはひと目見ればそうした箇所を発見します（そして，一瞬ではねられる）．

履歴書，履歴書ノートと並行する形で個人の日誌をつけている知人もいます．いわゆる「業務ノート」は，転職する際に会社に残す必要がありますが，個人の日誌は対象外です．もっとも，日誌に企業に属する内容（業務上，業務外を問わず）を記述してしまうと，個人の日誌として扱われないので注意が必要かもしれません（とくに引き抜かれた場合など，後でトラブルに巻き込まれないように業務日誌と個人日誌は区別して管理するべき）．

個人日誌は毎日つける日記のようなものではありません．プロジェクトの中で感じたこと（とくにミスやトラブルが発生したとき）や発見したことなどを，簡単にメモしておくといったたぐいのものです．履歴書ノートをまとめる際に役立ちますし，自分を振り返るときの資料にもなります．

インタビュー模擬体験

答 前回の解答（2006年2月号，pp.88-89）

問題

自分のプロジェクト・チームに，アウトソーシング（外部の協力会社）の人材を取り入れます．このとき，アウトソーシングした人材をどのようにマネージメントしていくか，プロジェクト・リーダの立場で説明してください．

プロジェクト・リーダとして，まずはアウトソーシングをどう位置づけるのかを明確にする必要があります．アウトソーシングする会社を自由に選択できればよいのですが，会社（あるいは部，課）によっては，包括契約などの理由でアウトソーシング先が固定される場合もあります．

いずれにしても，次の項目は押さえておきましょう．どれも初歩的なことばかりです．

- （アウトソーシング先の）だれが自分のプロジェクトに従事するのか，その人材は掛け持ちなのかフルタイムなのか，人材をローテーションするのかどうかなどを確認する．その人材に関する履歴書を確認することもたいせつ．（自分のプロジェクトにとっての）マネージャはだれなのかを明確にする（書面に残すこと）．少なくとも，一度はメンバと顔合わせをして面識を作ることも重要．ネットワーク時代とは言え，プロジェクトは人が行うもの．電話や電子メールだけでの知り合いというのはきずなが弱い．
- （アウトソーシング先の）シニア・レベルのマネージメントがどれだけ確かなのかを十分に確認する．自分が係長レベルのプロジェクト・マネージャなら，部長や課長同伴で相手方のシニア・レベルのマネージャとミーティングするとよい．
- プロジェクトの工程表，仕様，段取り（例えば，共同でデバッグするとき，別々の場所で実施するのか，アウトソーシング側がこちらに常駐する形で実施するのかなど）をプロジェクトを開始する前に十分煮詰め，確認する（書面にしておく）．はじめに段取りのレベルまで検討しないケースがよく見られるが，「どの段階で中間レビューを実施する」，「インテグレーションは共同で（同じフロアで）実施する」など，最初からある程度詳しく確認しておくと，後々のトラブルを未然に防げる．
- 海外のアウトソーシングを使う場合に顕著だが，相手方の会社のカルチャ（海外であればその国の文化や習慣など）や組織のしくみを十分に理解しておく必要がある．時間の概念一つとっても，自分の会社（国，社会）とかなり異なるのが実情である．

実際にアウトソーシングを使ってプロジェクトを進める場合，コミュニケーション・ミスマッチ（同じ単語でも意味することが異なったりする）からくるトラブルの発生は日常茶飯事です．下請けだから無理が利くなど，「元請け」，「下請け」といった上下関係で物事を考えるとうまくいきません．

アウトソーシングについては関連書籍がたくさんありますが，経験するのがいちばんです．今はプロジェクト・マネージャの立場でなくても，例えばマネージャを補助する形でアウトソーシングとかかわりを持つことは，キャリアアップに欠かせないと思います．

問 今月の問題（今月は技術的なエクササイズです．）

問題

次の記述について，おかしな点を指摘してください．

(1)
```
#include <stdio.h>
#include <stdlib.h>
int main()
{
  char name[50];
  FILE *infile;
  printf("Type the filename: ");
  fgets(name, sizeof(name), stdin);
  infile = fopen(name, "r");
  if (infile == NULL) {
    printf("ERROR¥n");
    exit(1);
  }
  fclose(infile);
}
```

(2)
```
int table[30][40];
#define init_table()
        memset(table, 1, sizeof(table));
```

(3)
```
#perl
  $x='me';
  $y='you';
  print "is x and y equal?",
               $x==$y, "¥n";
```

ハードウェア・エンジニア，ソフトウェア・エンジニアを問わず，スクリプト言語に通じていることが望まれます．さらに，ハードウェア・エンジニアはC言語でプログラミングができることがプラスになります．その背景として，C言語やPerlを駆使してテストベンチ環境を設計するのが一般的になっていることが挙げられます．

今回の問題は非常に簡単です．Perlを使って理解不可能なプログラムを作るマニアックなエンジニアもいますが，人に理解されないプログラムはだれからも好かれません．

とうはら・ともなり